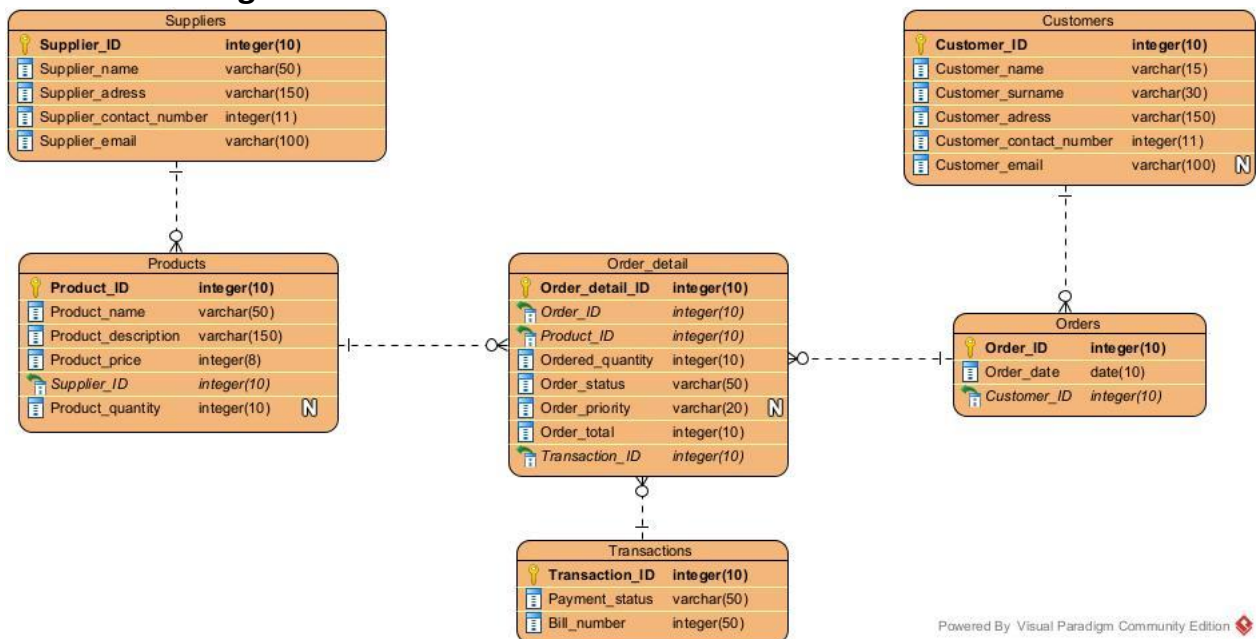


Inventory Management Database in Clothing Shop

1. Business topic and description

Effective inventory management is the key to success for any clothing shop. Our inventory database optimizes operations by tracking stock levels and purchase history. Real-time updates enable quick responses to changing demands, reducing overstock and stockouts. This system maximizes customer satisfaction and ensures product availability while also supporting data-driven decision-making and improving operational efficiency. An inventory management database includes multiple tables to organize and store different aspects of inventory-related information.

2. Data diagram



3. Functions:

- Product Management:

Allows users to add, edit, or remove products from the inventory.

- Order Processing:

Facilitates the management of supplier orders and updates inventory levels accordingly.

- Stock Tracking and Monitoring:

Provides real-time visibility into stock levels and alerts when inventory falls below specified reorder points.

- Sales and Order Fulfillment:

Records sales transactions, updates stock levels, and manages order fulfillment processes.

- Location Management:

Tracks and manages inventory stored in different locations or warehouses.

- Returns and Refunds:

Manages the processing of product returns, updates inventory levels, and handles refunds.

- Supplier Relationship Management (SRM):

manage lead times effectively, and enhance overall supply chain efficiency.

- Reporting and Analytics:

Generates reports and analytics to provide insights into inventory performance, trends, and stock turnover.

4. Views:

1. Paid Orders View:

This view filters and displays only those orders where the payment has been successfully processed. It helps in identifying and managing orders that have been paid for, streamlining the fulfillment process.

```
/* create view view1 as (  
select*  
from transactions  
where payment_status = 'PAID'; */
```

2. Quantity Ascending View:

This view arranges products in descending order based on their quantities in stock. It provides a quick overview of the inventory, highlighting which products have the highest stock levels and which may need attention, such as reordering or marketing promotions.

```
/* create view view2 as (  
select product_quantity, product_name, product_ID from products)*/
```

after creating view:

```
select * from view2  
order by product_quantity asc;
```

3. Which Supplier Delivers Which Product View:

This view establishes a link between the suppliers and the products they deliver. It's useful for tracking the supplier associated with each product, streamlining communication with suppliers, and assessing the performance of each supplier in terms of product delivery.

```
/*create view view3 as (  
  select supplier_id, supplier_name, product_id, product_name, product_quantity  
  from suppliers join products using ( supplier_id)); */
```

4. Transaction Type: Cancelled View:

This view filters specifically cancelled transactions. It allows to monitor them easily and also provides management of cancelled transactions, helping to track them, and make sure that cancelled orders do not get fulfilled.

```
/*create view view4 as (  
  select *  
  from transactions  
  where payment_status = 'CANCELLED'  
);*/
```

5. Priority View:

This view shows orders that are prioritized. Orders get prioritised based on urgency, possibly considering factors like order types, delivery deadlines, or customer preferences. It assists in managing workload and ensuring that high-priority orders are processed promptly.

```
/*create view view5 as (  
  select order_priority, order_details.order_id  
  from order_details  
  join orders on order_details.order_id = orders.order_id  
  where order_priority is not null);*/
```

6. Sort by Order Date (From the Oldest) View:

This view organizes orders based on their order dates in their order. There is a possibility to order dates for example displaying the oldest orders first. It provides a historical perspective and aids in managing and fulfilling orders in a chronological fashion.

```
/*create view view6 as (  
  select order_id, order_date  
  from orders);*/
```

to display in certain order:

ascending: select * from view6 order by order_date asc

descending: select * from view6 order by order_date desc

7. Quantity Below 20 View:

This view filters products with quantities below a threshold, specifically 20 in this case. It helps in identifying products that are running low in stock, triggering timely reorder decisions to prevent stockouts.

```
/*create view view7 as(
  select*
  from products
  where product_quantity < 20); */
```

8. Customer Name, Surname – Order ID View:

This view combines customer names, surnames, and order IDs. It's useful for customer service and order tracking, making it easy to locate specific orders and communicate with customers regarding their purchases.

```
/*create view view8 as (
select customer_name, customer_surname, order_id
from customers join orders using (customer_id))*/
```

SQL Worksheet

```
/*CREATE TABLE orders(
  order_id VARCHAR(10) PRIMARY KEY,
  order_date DATE NOT NULL,
  customer_id VARCHAR(10) NOT NULL
);
```

```
INSERT INTO orders (order_id, order_date, customer_id)
VALUES ('Z001', TO_DATE('2023-12-29', 'YYYY-MM-DD'), 'niweu1234');
INSERT INTO orders (order_id, order_date, customer_id)
VALUES ('Z002', to_date('2023-12-29', 'YYYY-MM-DD'), 'bank123');
INSERT INTO orders (order_id, order_date, customer_id)
VALUES ('Z003', to_date('2023-12-30', 'YYYY-MM-DD'), 'njui345');
INSERT INTO orders (order_id, order_date, customer_id)
VALUES ('Z004', to_date('2024-01-01', 'YYYY-MM-DD'), 'jhkg123');
INSERT INTO orders (order_id, order_date, customer_id)
VALUES ('Z005', to_date('2024-01-02', 'YYYY-MM-DD'), 'jkoi390');
INSERT INTO orders (order_id, order_date, customer_id)
VALUES ('Z006', to_date('2024-01-04', 'YYYY-MM-DD'), 'neio8978');
```

```
CREATE TABLE suppliers(
  supplier_id VARCHAR(10) PRIMARY KEY,
  supplier_name VARCHAR(50) NOT NULL,
  supplier_adress VARCHAR(150) NOT NULL,
```

Anastasiia Skrypnyk

```
supplier_contact_number NUMBER(11) NOT NULL,  
supplier_email VARCHAR(100) NOT NULL  
);
```

```
INSERT INTO suppliers (supplier_id, supplier_name, supplier_adress,  
supplier_contact_number , supplier_email)  
VALUES ('2023_1', 'Amore', 'Aleja Krakowska 27/27, 05-090 Sękocin Nowy – Poland',  
48576712876, 'amoreclothing@gmail.com');  
INSERT INTO suppliers (supplier_id, supplier_name, supplier_adress,  
supplier_contact_number , supplier_email)  
VALUES ('2023_2', 'Makoma', 'Dobrzyńska 73, 87-811 Włocławek – Poland', 48678543908,  
'makoma@gmail.com');  
INSERT INTO suppliers (supplier_id, supplier_name, supplier_adress,  
supplier_contact_number , supplier_email)  
VALUES ('2023_3', 'Kamus', 'Sikorskiego 12a/8, 70-313 Szczecin - Poland', 48765987123,  
'kamusfirma@gmail.com');  
INSERT INTO suppliers (supplier_id, supplier_name, supplier_adress,  
supplier_contact_number , supplier_email)  
VALUES ('2023_4', 'M-crop', 'Orchow 176, 98-100 Lask - Poland', 48654900432,  
'm.crop@gmail.com');  
INSERT INTO suppliers (supplier_id, supplier_name, supplier_adress,  
supplier_contact_number , supplier_email)  
VALUES ('2023_5', 'Kapabol', 'Willowa 9, 05-410 Józefów - Poland', 48433765912,  
'kapabol@gmail.com');
```

```
CREATE TABLE products(  
product_id VARCHAR(10) PRIMARY KEY,  
product_name VARCHAR(50) NOT NULL,  
product_description VARCHAR(150) NOT NULL,  
product_price NUMBER(8) NOT NULL,  
product_quantity NUMBER(10),  
supplier_id VARCHAR(10),  
CONSTRAINT fk_suppliers  
FOREIGN KEY(supplier_id)  
REFERENCES suppliers(supplier_id)  
);
```

```
INSERT INTO products(product_id, product_name, product_description, product_price,  
product_quantity, supplier_id)  
VALUES ('001', 'Basic T-shirt', 'Cotton T-shirt; Colours: White, Black, Brown; Sizes: XS, S, M,  
XL', 50.00, 300, '2023_4');  
INSERT INTO products(product_id, product_name, product_description, product_price,  
product_quantity, supplier_id)  
VALUES ('002', 'Long Sleeve T-shirt', 'Cotton T-shirt; Colours: White, Black, Grey; Sizes: XS, S,  
M, XL', 60.00, 150, '2023_4');
```

```
INSERT INTO products(product_id, product_name, product_description, product_price,
product_quantity, supplier_id)
VALUES ('003', 'Neck Bodysuit', 'Polyester Bodysuit; Colours: White, Black, Blue, Green. Pink;
Sizes: S, M, L', 30.00, 10, '2023_2');
INSERT INTO products(product_id, product_name, product_description, product_price,
product_quantity, supplier_id)
VALUES ('004', 'Short Sleeve Crop Top', 'Polyester Crop Top; Colours: White, Black, Orange,
Green; Sizes: XS, S, M, L', 30.00, 100, '2023_2');
INSERT INTO products(product_id, product_name, product_description, product_price,
product_quantity, supplier_id)
VALUES ('005', 'Satin Halter Top', 'Polyester Top; Colours: White, Black; Sizes: XS, S, M, L',
40.00, 20, '2023_2');
INSERT INTO products(product_id, product_name, product_description, product_price,
product_quantity, supplier_id)
VALUES ('006', 'Maxi Sleeve Dress', 'Modal Dress; Colours: White, Black, Grey, Pink; Sizes: XS,
S, M, L', 90.00, 70, '2023_1');
INSERT INTO products(product_id, product_name, product_description, product_price,
product_quantity, supplier_id)
VALUES ('007', 'Mini Sleeve Dress', 'Modal Dress; Colours: White, Black, Grey, Pink; Sizes: XS,
S, M, L', 90.00, 15, '2023_1');
INSERT INTO products(product_id, product_name, product_description, product_price,
product_quantity, supplier_id)
VALUES ('008', 'Maxi Satin Dress', 'Satin Dress; Colours: White, Black; Sizes: XS, S, M, L',
100.00, 70, '2023_1');
INSERT INTO products(product_id, product_name, product_description, product_price,
product_quantity, supplier_id)
VALUES ('009', 'High-Waist Mini Skirt', 'Modal Skirt; Colours: White, Black; Sizes: XS, S, M, L',
60.00, 50, '2023_3');
INSERT INTO products(product_id, product_name, product_description, product_price,
product_quantity, supplier_id)
VALUES ('010', 'Maxi Jeans Skirt', 'Jeans Skirt; Colours: Black, Blue; Sizes: XS, S, M, L', 70.00,
14, '2023_3');
INSERT INTO products(product_id, product_name, product_description, product_price,
product_quantity, supplier_id)
VALUES ('011', 'Maxi Satin Skirt', 'Satin Skirt; Colours: White, Black; Sizes: XS, S, M, L', 60.00,
40, '2023_3');
INSERT INTO products(product_id, product_name, product_description, product_price,
product_quantity, supplier_id)
VALUES ('012', 'High-Rise Straight Jeans', 'Cotton Jeans; Colours: White, Black, Blue; Sizes: XS,
S, M, L', 90.00, 100, '2023_5');
INSERT INTO products(product_id, product_name, product_description, product_price,
product_quantity, supplier_id)
VALUES ('013', 'High-Rise Skinny Jeans', 'Cotton Jeans; Colours: White, Black, Blue; Sizes: XS,
S, M, L', 80.00, 30, '2023_5');
INSERT INTO products(product_id, product_name, product_description, product_price,
product_quantity, supplier_id)
```

```
VALUES ('014', 'Wide-Leg Jeans', 'Cotton Jeans; Colours: White, Black, Blue; Sizes: XS, S, M, L',  
90.00, 50, '2023_5');
```

```
CREATE TABLE customers (  
    customer_ID VARCHAR(10) PRIMARY KEY,  
    customer_name VARCHAR(15) NOT NULL,  
    customer_surname VARCHAR(30) NOT NULL,  
    customer_address VARCHAR(150) NOT NULL,  
    customer_contact_number NUMBER(10) NOT NULL,  
    customer_email VARCHAR(100)  
);
```

```
INSERT INTO customers ( customer_ID, customer_name, customer_surname,  
customer_address, customer_contact_number,customer_email)  
VALUES ('niweu1234', 'Alice', 'Boston', 'Kamienna 190/23, 50-545 Wroclaw',678876567,  
'aliceboston@gmail.com');  
INSERT INTO customers ( customer_ID, customer_name, customer_surname,  
customer_address, customer_contact_number,customer_email)  
VALUES ('bank123', 'Margaret', 'Flower', 'Stone 2, 69-908 London',  
909876543, 'margaret@gmail.com');  
INSERT INTO customers ( customer_ID, customer_name, customer_surname,  
customer_address, customer_contact_number,customer_email)  
VALUES ('njud345', 'Agata', 'Byk', 'Zlota 2, 43-098 Warszawa',  
8765443123, 'agatka@gmail.com');  
INSERT INTO customers ( customer_ID, customer_name, customer_surname,  
customer_address, customer_contact_number,customer_email)  
VALUES ('jhkg123', 'Marcin', 'Cukier', 'Modrzewiowa 1/9, 32-908 Krakow', 787787989,  
'marcincukier@gmail.com');  
INSERT INTO customers ( customer_ID, customer_name, customer_surname,  
customer_address, customer_contact_number,customer_email)  
VALUES('jkoi390', 'Jadwiga', 'Kot', 'Partyzantow 3, 59-900 Gy', 786543425, 'jadzia@gmail.com');  
INSERT INTO customers ( customer_ID, customer_name, customer_surname,  
customer_address, customer_contact_number,customer_email)  
VALUES('neio8978', 'Olaf', 'Snow', 'Lodowa 76/12, 89-900 Gdansk', 766552414, null);
```

```
CREATE TABLE transactions(  
    transaction_ID NUMBER(10) PRIMARY KEY,  
    payment_status VARCHAR(30) NOT NULL,  
    bill_number NUMBER(30) NOT NULL  
);
```

```
INSERT INTO transactions (transaction_ID, payment_status, bill_number)  
VALUES (6578532, 'PAID',543278193336);  
INSERT INTO transactions (transaction_ID, payment_status, bill_number)  
VALUES(876240, 'OPEN', 7623354);
```

Anastasiia Skrypnyk

```
INSERT INTO transactions (transaction_ID, payment_status, bill_number)
VALUES (0823698,'COMPLETE',98726391);
INSERT INTO transactions (transaction_ID, payment_status, bill_number)
VALUES(6180903,'CANCELLED',203863);
INSERT INTO transactions (transaction_ID, payment_status, bill_number)
VALUES (3275312,'OPEN', 36632683);
INSERT INTO transactions (transaction_ID, payment_status, bill_number)
VALUES (2863192,'COMPLETE',7379320);
```

```
INSERT INTO transactions (transaction_ID, payment_status, bill_number)
VALUES(0239903,'COMPLETE',032022);
INSERT INTO transactions (transaction_ID, payment_status, bill_number)
VALUES(038320,'CANCELLED',3081000);
```

```
CREATE TABLE order_details(
  order_detail_id VARCHAR(10) NOT NULL,
  order_id VARCHAR(10) NOT NULL,
  CONSTRAINT fk_orders
    FOREIGN KEY(order_id)
    REFERENCES orders(order_id),
  product_id VARCHAR(10) NOT NULL,
  CONSTRAINT fk_products
    FOREIGN KEY(product_id)
    REFERENCES products(product_id),
  ordered_quantity INTEGER NOT NULL,
  order_status VARCHAR(50) NOT NULL,
  order_priority VARCHAR(20),
  order_total INTEGER NOT NULL,
  transaction_id NUMBER(10) NOT NULL,
  CONSTRAINT fk_transactions
    FOREIGN KEY(transaction_id)
    REFERENCES transactions(transaction_id)
);
```

```
INSERT INTO order_details (order_detail_id, order_id, product_id, ordered_quantity,
order_status, order_priority, order_total, transaction_id)
VALUES ('DZ001', 'Z001', '006', 3, 'SENT', null, 270.00, 6578532);
```

```
INSERT INTO order_details (order_detail_id, order_id, product_id, ordered_quantity,
order_status, order_priority, order_total, transaction_id)
VALUES ('DZ002', 'Z002', '011', 6, 'PENDING', 'high', 360.00, 6180903);
```

```
INSERT INTO order_details (order_detail_id, order_id, product_id, ordered_quantity,
order_status, order_priority, order_total, transaction_id)
VALUES ('DZ003', 'Z003', '001', 10, 'DELIVERED', null, 500.00, 0823698);
```


Anastasiia Skrypnyk

```
INSERT INTO order_details (order_detail_id, order_id, product_id, ordered_quantity,  
order_status, order_priority, order_total, transaction_id)  
VALUES ('DZ004', 'Z004', '003', 1, 'PENDING', 'medium', 30.00, 2863192);
```

```
INSERT INTO order_details (order_detail_id, order_id, product_id, ordered_quantity,  
order_status, order_priority, order_total, transaction_id)  
VALUES ('DZ005', 'Z005', '014', 2, 'PENDING', null, 180.00, 0239903);
```

```
INSERT INTO order_details (order_detail_id, order_id, product_id, ordered_quantity,  
order_status, order_priority, order_total, transaction_id)  
VALUES ('DZ006', 'Z006', '009', 5, 'PENDING', null, 300.00, 038320);*/
```

```
/* create view view1 as (  
select*  
from transactions  
where payment_status = 'PAID'; */
```

```
/* create view view2 as (  
    select product_quantity, product_name, product_ID from products)*/
```

```
/*select * from view2  
order by product_quantity asc;*/
```

```
/*create view view3 as (  
    select supplier_id, supplier_name, product_id, product_name, product_quantity  
    from suppliers join products using ( supplier_id));*/
```

```
/*create view view4 as (  
select *  
from transactions  
where payment_status = 'CANCELLED'  
);*/
```

```
/*create view view5 as (  
select order_priority, order_details.order_id  
from order_details  
join orders on order_details.order_id = orders.order_id  
where order_priority is not null);*/  
/*create view view6 as (  
select order_id, order_date  
from orders);*/
```

```
/*select * from view6 order by order_date asc*/  
/*select * from view6 order by order_date desc*/
```

Anastasiia Skrypnyk

```
/*create view view7 as(  
  select*  
  from products  
  where product_quantity < 20); */
```

```
/*create view view8 as (  
select customer_name, customer_surname, order_id  
from customers join orders using (customer_id))*/
```